Local features detectors play an important role in many applications like mapping, text recognition, image registration (J. Bauer et al., 2004), object recognition (A. Berg et al., 2005), object categorization (Dorko and Schmid, 2003), texture classification (S. Lazebnik et al., 2005), robot localization (S. Se et al., 2001), and video shot retrieval (J. Sivic et al., 2006). There are many researches that build new fast and robust detector (SIFT (D. Lowe, 2004), SURF (H. Bay et al., 2008), FAST (Guo, 2011), BRISK (Leutenegger, 2011) and descriptors SIFT (D. Lowe, 2004), SURF (H. Bay et al., 2008), BRISK (Leutenegger, 2011), Harris-C. Harris and M. Stephens (1988), FREAK (A. Alahi et al., 2012), MinEigen, MSER, HOG). Local features can be utilized into two different methods. First method includes three steps: feature detection, feature description, and feature matching (S. SriVidhya et al., 2015). Second method is bag-of-features (E. Nowak et al., 2006) and hyper features (Agarwal et al., 2006) that includes feature detection, feature description, feature clustering, and frequency histogram construction for image representation. A local feature extraction is composed of feature detector and a feature descriptor. In this paper, we have discussed about the performance of various features detectors such as FAST, MSER, SURF, Harris, and MinEigen.

2. Feature Detectors

2.1. Features From Accelerated Segment Test (FAST):
It was proposed originally by Rosten and Drummond (E. Rosten, and T. Drummond, 2006) for identifying interest points in an image. An interest point in an image is a pixel which has a well-defined position and can be robustly detected. Interest points have high local information content and they should be ideally repeatable between different images (Edward Rosten et al., 2010). It is proven that FAST detector performs well on images acquired by mobile devices in the context of visual navigation (Michal Nowicki and Riut, 2014). Interest point detection has applications in image matching, object recognition, tracking etc.

Segment test detector uses a circle of 16 pixels (a Bresenham circle of radius 3) to classify whether a candidate point is actually a corner. Each pixel in the circle is labelled from integer number 1 to 3. The four example pixels of radius 3 have been shown in Figure 1. If a set of N contiguous pixels in the circle are all brighter than the intensity of candidate pixel p (denoted by I_p) plus a threshold value t or all darker than the intensity of candidate pixel p minus threshold value t, then p is classified as corner.

An interest point in an image is a pixel which has a well-defined position and can be robustly detected.

2.2. Maximally Stable External Regions (MSER):
MSERs (O. Chum and J. Matas, 2005) are regions that are either darker or brighter than their surroundings, and that are stable across a range of thresholds of the intensity function. MSERs have also been defined on color scalar functions (S. Obdrzalek, 2007), and have been extended to color (P. E. Forssen, 2007). This is used to detect blobs in an image. This technique was proposed by Matas et al. (2002) to find correspondences between image elements from two images with different viewpoints. This method of extracting a comprehensive number of corresponding image elements contribute to the wide-baseline matching, and it has led to better stereo matching and object recognition algorithms. MSER can efficiently extract crosswalk regions under various illumination conditions, which can avoid the selection of thresholds according to the current environment situation and greatly improve the system flexibility and robustness (Yuqiang Zhai et al., 2015).

2.3. Speeded-Up Robust Features (SURF):
It was proposed by Herbert Bay et al. at European Conference on Computer Vision (Ryuji et al., 2009). It is a local feature detector and descriptor that can be used for tasks such as object recognition.
2.4. Harris Detector:
Harris and Stephens (C. Harris and M. Stephens, 1988), improved upon Moravec’s corner detector by considering the differential of the corner score with respect to direction directly, instead of using shifted patches. This corner score is referred to as autocorrelation. Without loss of generality, they have assumed a gray scale 2-dimensional image. Let this image be given by I. Consider taking an image patch over the area \( (u,v) \) and shifting it by \( (u,v) \). The weighted sum of squared differences (SSD) between these two patches, denoted \( S \), is given by:

\[
S(x,y) = \sum_{u,v} w(u,v)(I(x+u,y+v) - I(x,y))^2
\]

This produces the approximation

\[
S(x,y) = \sum_{u,v} w(u,v)I_x(x,y + I_y(x,y))
\]

This can be written in matrix form:

\[
S(x,y) = (x,y) A (x,y)^T
\]

where \( A \) is the structure tensor,

\[
A = \sum w(u,v) \begin{bmatrix} I_x & I_y \\ I_y & I_z \end{bmatrix} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}
\]

This matrix is a Harris matrix, and angle brackets denote averaging. A corner is characterized by a large variation of \( S \) in all directions of the vector \( (x,y) \). By analyzing the eigenvalues of \( A \), this characterization can be expressed in the following way: \( A \) should have two “large” eigenvalues for a corner. Based on the magnitudes of the eigenvalues, the following inferences can be made based on this argument:

1. If \( \lambda_1 = 0 \) and \( \lambda_2 = 0 \) then this pixel \( (x,y) \) has no features of interest.
2. If \( \lambda_1 > 0 \) and \( \lambda_2 > 0 \) has some large positive value, then an edge is found.
3. If \( \lambda_1 \) and \( \lambda_2 \) have large positive values, then a corner is found.

2.5. MinEigen:
Detect corners using minimum eigenvalue algorithm and return corner Points Object. The object contains information about the feature points detected in a 2-D gray scale input image, I. The detectMinFeatures method in Matlab uses the minimum eigenvalue algorithm developed by Shi and Tomasi to find feature points (Shi, J., & Tomasi, C., 1994).

3. Experiments
This paper aims to evaluate the various feature detection algorithms. The implementation was done on Intel® Core(TM) i3 processor with 3GB RAM and speed of 2.53GHz. The code was written in Matlab R2013a on Windows 7 professional 64 bits. It consists of various tests by introducing effects like rotation, scale change and noise. The sample picture considered for all the tests is shown below in Fig 3.1 of size 35.2KB.

First, we will evaluate the detectors by number of captured key-points against elapsed time with rotational changes of 25, 45, 75, and 100 in an image. The following table’s shows number of detected key points and time needed to capture them.

<table>
<thead>
<tr>
<th>Detectors</th>
<th>Detected feature points</th>
<th>Elapsed time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original image</td>
<td>Rotate 25</td>
<td>Rotate 45</td>
</tr>
<tr>
<td>FAST</td>
<td>223</td>
<td>192</td>
</tr>
<tr>
<td>MSER</td>
<td>130</td>
<td>102</td>
</tr>
<tr>
<td>SURF</td>
<td>150</td>
<td>156</td>
</tr>
<tr>
<td>Harris</td>
<td>163</td>
<td>194</td>
</tr>
<tr>
<td>MinEigen</td>
<td>563</td>
<td>734</td>
</tr>
</tbody>
</table>

Second, and in order to evaluate the detectors by number of captured key-points against elapsed time with scale changes of 1.2, 1.5, 1.7, and 1.9 in an image. The following table’s shows number of detected key points and time needed to capture them.

<table>
<thead>
<tr>
<th>Detectors</th>
<th>Detected feature points</th>
<th>Elapsed time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original image</td>
<td>Scale 1.2</td>
<td>Scale 1.5</td>
</tr>
<tr>
<td>FAST</td>
<td>223</td>
<td>219</td>
</tr>
<tr>
<td>MSER</td>
<td>130</td>
<td>156</td>
</tr>
<tr>
<td>SURF</td>
<td>150</td>
<td>214</td>
</tr>
<tr>
<td>Harris</td>
<td>163</td>
<td>216</td>
</tr>
<tr>
<td>MinEigen</td>
<td>563</td>
<td>792</td>
</tr>
</tbody>
</table>

Third, and in order to evaluate the detectors by number of captured key-points against elapsed time with introducing various types of noises like Gaussian, Poisson, Salt & Pepper in an image. The following table shows number of detected key points and time needed to capture them.

<table>
<thead>
<tr>
<th>Detectors</th>
<th>Detected feature points</th>
<th>Elapsed time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original image</td>
<td>Gaussian</td>
<td>Poisson</td>
</tr>
<tr>
<td>FAST</td>
<td>223</td>
<td>776</td>
</tr>
<tr>
<td>MSER</td>
<td>130</td>
<td>152</td>
</tr>
<tr>
<td>SURF</td>
<td>150</td>
<td>194</td>
</tr>
<tr>
<td>Harris</td>
<td>163</td>
<td>968</td>
</tr>
<tr>
<td>MinEigen</td>
<td>563</td>
<td>1432</td>
</tr>
</tbody>
</table>

4. Conclusion:
The main purpose of this paper is to find the best detector in terms of rotation, scale change and various noises. We have deduced from the performance analysis that Min Eigen is the ideal choice amongst FAST, SURF, MSER, Harris feature detector algorithms. Min Eigen detects more features and provides better results, even under the rotational, scaling changes and introduction of noise. As...
shown in performance evaluation tables above, we see that FAST is taking less time but the number of features detected is less. Min Eigen takes marginally high time when compared to FAST but provides better results (time difference — scale: 0.015, rotation: 0.068, Noise: 0.106). Thus ignoring this marginal increase we suggest that Min Eigen is the optimal feature detection algorithm which can be used efficiently used in the SLAM procedure. We are planning to combine these detectors and descriptors to match the image precisely in order to attain better feature extraction results.

References: